(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 November 2003 (13.11.2003)

PCT

(10) International Publication Number
WO 03/093979 A1

(51) International Patent Classification[7]: G06F 9/35, 9/45

(21) International Application Number: PCT/EP02/04927

(22) International Filing Date: 3 May 2002 (03.05.2002)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **ANTEVISTA GMBH** [DE/DE]; Harburger Schlossstrasse 6-12, 21079 Hamburg (DE).

(71) Applicant and
(72) Inventor: **THEIS, Jean-Paul** [LU/LU]; 1, Porte des Ardennes, L-9145 Erpeldange (LU).

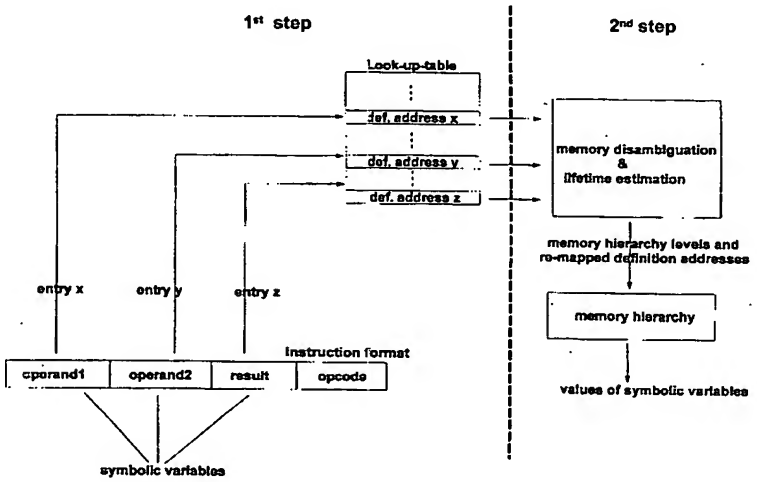
(81) Designated States *(national)*: JP, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** A METHOD FOR REALIZING AUTONOMOUS LOAD/STORE BY USING SYMBOLIC MACHINE CODE



(57) **Abstract:** The invention describes a method for realizing autonomous load/store by using symbolic machine code. Instructions in symbolic machine code may have operands and/or results specifying symbolic variables. Symbolic variables naturally arise as the machine code pendant of pointer variables declared in a program written in some high level programming language. Formally, a symbolic variable specifies an entry in a dedicated memory other than a register file of said microprocessor. Said entry is used by the microprocessor in order to determine the definition address of said variable. By analyzing the symbolic variables specified as instruction operands and results and by writing the definition addresses of said symbolic variables into a heap address cache, the microprocessor dynamically determines the lifetimes of the values of said variables and, based on the lifetimes, determines where in the memory hierarchy said values will be stored.

WO 03/093979 A1